2. Напечатать первые 𝑁 простых чисел

Просто́е число́ — натуральное число, имеющее ровно два различных натуральных делителя. Другими словами, натуральное число p является простым, если оно отлично от 1 и делится без остатка только на 1 и на само p.

#include <iostream>  
#include <cmath>  
using namespace std;

int main() {

int N=10;  
int k = 0;  
for (int i = 2; k < N; i++){  
bool fl = true;  
for (int j = 2; j <= sqrt(double(i)); j++)  
if (i % j == 0){  
fl = false;  
break; }  
if (fl ) {  
cout << i << " ";  
k++; }}}

8. Напечатать квадраты всех целых чисел от 0 до 𝑁, используя только операции сложения и вычитания, не используя вложенных циклов.  
int main() {  
int N=10;  
int xs = 0;  
for (int x = 0; x < N; x++){//(x+1)\*(x+1) = x\*x + x + x + 1  
xs = xs + x + x + 1;  
cout << xs << " "; }  
cout << endl;  
int x = 0, i = 1, k = 0; // или  
for(; k < N; i += 2){//разность м/у квадратами увеличивается на два  
x += i;  
//cout << x << " ";  
k++; } }

12. Вычислить значения функции в диапазоне [𝑎, 𝑏] c шагом 𝑕. Если функция неопределена, вывести сообщение типа «Деление на ноль», «Логарифм из неположительного числа», «Корень из отрицательного числа»:



#include <iostream>  
#include <cmath>  
using namespace std;  
int main() {  
double x;//12 если не исследован корень - не ошибка  
cin>> x;  
if (x == 5) cout << " Деление на ноль\n";  
else if (x < 1 && x > -1) cout << " Корень из отр. числа и логарифм из непол. числа\n";  
else if (x <= 2) cout << " Логарифм из неположительного числа\n";  
else cout << (sqrt(x \* x - 1) + log(x \* x \* x - 8))/(1 - exp(x - 5))  
<< endl; }

18. Вывести на экран следующую таблицу:

1

9

1 2

9 7

1 2 3

9 7 5

1 2 3 4

9 7 5 3

#include <iostream>  
#include <cmath>  
using namespace std;  
int main()  
{  
for(int i = 0; i < 4; i++){  
for (int j = 0; j <= i; j++)  
cout << j + 1 << " ";  
cout << endl;  
for(int j = 0; j <= i; j++)  
cout << 9 - 2 \* j << " ";  
cout << endl;  
}  
}